Google Play Store Apk Bundle ((TOP))

Google Play Store APK Bundle: What You Need to Know

If you are an Android user or developer, you might have heard about the recent change in the Google Play Store. Starting from August 2021, new apps are required to publish with the Android App Bundle (AAB) instead of the Android Application Package (APK) as the standard publishing format. This change affects how apps are built, uploaded, downloaded, and installed on Android devices. In this article, we will explain what an AAB is, why Google is making this change, how it benefits users and developers, and how you can publish your app with AAB.

Introduction

What is an Android App Bundle (AAB)?

An Android App Bundle is a publishing format that includes all your app's compiled code and resources, and defers APK generation and signing to Google Play. Google Play uses your app bundle to generate and serve optimized APKs for each device configuration, so only the code and resources that are needed for a specific device are downloaded to run your app. You no longer have to build, sign, and manage multiple APKs to optimize support for different devices, and users get smaller, more-optimized downloads.

Why is Google replacing APK with AAB?

Google is replacing APK with AAB to provide a modern Android distribution that benefits all developers and users. AAB enables more efficient app delivery and updates, reduces device storage usage, improves app performance and security, and supports advanced features such as dynamic delivery and asset delivery. Google claims that over a million apps are already using AAB format, including popular apps such as Netflix, Twitter, Adobe, and Duolingo.

How does AAB benefit users and developers?

AAB benefits users by providing them with smaller app downloads and updates, faster installation speed, less device storage consumption, better app performance and security, and more customized app experiences based on their device specifications and preferences. AAB benefits developers by simplifying the app development and publishing process, reducing the complexity of managing multiple APKs, increasing the number of app installs and retention rates, enabling modular app development with feature modules and asset packs, and accessing new performance features available on newer devices.

Main Features of AAB

Support for various device configurations and languages

AAB supports various device configurations such as screen size, resolution, density, CPU architecture, RAM size, etc., as well as different languages. Google Play generates optimized APKs for each device configuration based on the app bundle's code and resources. This means that users only download the code and resources that match their device's specifications and settings. For example, if a user has a device with an ARMv8 CPU architecture and a French language setting, they will only download the APK that contains the ARMv8 native libraries and the French strings.

Smaller app size and faster download speed

AAB reduces the app size by eliminating unnecessary code and resources from the APKs that are downloaded to the devices. According to Google's data, apps using AAB format are on average 15% smaller than apps using APK format. Smaller app size means faster download speed for users and less device storage consumption. Users are more likely to install and keep apps that are smaller and faster to download.

Increased number of installations and reduced number of uninstallations

AAB increases the number of app installations by reaching more users across different devices and regions. Google Play automatically splits your app bundle into APKs that are suitable for each device's configuration and language. This means that you don't have to worry about creating and maintaining multiple APKs for different devices and markets. You can also use the Google Play Console to manage your app's distribution and target specific segments of users based on their device features, country, or custom criteria. AAB also reduces the number of app uninstallations by improving the user experience and satisfaction. Users are less likely to uninstall apps that are optimized for their devices, run smoothly, and consume less storage space.

Play Feature Delivery and Play Asset Delivery

AAB enables two advanced features that allow you to deliver dynamic content to your users: Play Feature Delivery and Play Asset Delivery. Play Feature Delivery lets you deliver features on demand or conditionally, depending on the user's device capabilities, preferences, or country. You can use feature modules to create modular apps that can be installed, updated, or removed independently from the base app. For example, you can create a feature module for a game level that is only downloaded when the user reaches that level, or a feature module for a premium service that is only available in certain countries. Play Asset Delivery lets you deliver large assets more efficiently and with lower latency. You can use asset packs to bundle and compress your app's assets and deliver them through Google Play's high-performance CDN. You can also choose from three delivery modes: install-time, fast-follow, or on-demand, depending on when you want your assets to be available to your users. For example, you can use install-time delivery for essential assets that are needed for your app to start, fast-follow delivery for large assets that are needed shortly after the first app launch, or on-demand delivery for optional or rarely used assets that are requested by the user at runtime.

How to Publish Your App with AAB

Requirements and restrictions for AAB

To publish your app with AAB, you need to meet some requirements and follow some restrictions. The main requirements are: - You need to use Android Studio 3.2 or higher to build your app bundle. - You need to enroll in Play App Signing to let Google Play sign your app bundle and manage your app signing key. - You need to target API level 30 or higher for new apps and API level 28 or higher for updates. - You need to use bundletool to test your app bundle locally or on Firebase Test Lab. The main restrictions are: - You cannot publish instant apps with AAB format. - You cannot publish multiple APKs for the same app with AAB format. - You cannot publish APKs larger than 150 MB with AAB format. - You cannot use expansion files (OBB) with AAB format.

Steps to build and upload an AAB file

The steps to build and upload an AAB file are: - In Android Studio, select Build > Generate Signed Bundle/APK from the menu bar. - In the Generate Signed Bundle or APK dialog, select Android App Bundle and click Next. - In the Module dropdown menu, select the module you want to bundle. - In the Key store path field, select an existing key store or create a new one. - Enter the key store password, key alias, key password, and validity and certificate information. - Click Next. - In the Destination Folder field, select where you want to save your app bundle file (.aab). - Click Finish. - In the Google Play Console, go to the App bundle explorer page under Release > Setup > App integrity. - Click Upload new app bundle under App bundles in testing. - Select your app bundle file (.aab) from your computer and click Upload.

Tools and resources to help you with AAB

There are some tools and resources that can help you with AAB. Some of them are: - Bundletool: a command-line tool that lets you test how Google Play generates APKs from your app bundle. - Android Studio: an IDE that lets you build, test, debug, and optimize your app bundle. - Google Play Console: a web interface that lets you manage your app's distribution, performance, quality, and monetization. - Android App Bundle documentation: a collection of guides, tutorials, reference pages, and videos that explain how to use AAB format.

Conclusion

In conclusion, Google Play Store Store APK Bundle is a new publishing format that replaces the traditional APK format. It offers many benefits for users and developers, such as smaller app size, faster download speed, more device compatibility, more language support, more dynamic content delivery, and more modular app development. To publish your app with AAB, you need to use Android Studio, enroll in Play App Signing, target the latest API level, and use bundletool to test your app bundle. You can also use the Google Play Console and the Android App Bundle documentation to help you with AAB. If you are an Android user or developer, you should definitely check out the Google Play Store APK Bundle and see how it can improve your app experience.

FAQs

What is the difference between APK and AAB?

APK is the traditional publishing format for Android apps that contains all the code and resources for a single app variant. AAB is the new publishing format for Android apps that contains all the code and resources for multiple app variants. Google Play uses AAB to generate and serve optimized APKs for each device configuration.

How can I download an AAB file from Google Play?

You cannot download an AAB file from Google Play directly. Google Play only serves APK files to your device based on your device configuration and language. If you want to download an AAB file, you need to contact the app developer and ask them to provide you with the AAB file.

How can I install an AAB file on my device?

You cannot install an AAB file on your device directly. You need to use a tool called bundletool to convert the AAB file into APK files and then install them on your device. You can find bundletool on

GitHub or download it from Android Studio.

How can I update my app with AAB format?

You can update your app with AAB format the same way you update your app with APK format. You just need to upload a new version of your app bundle to the Google Play Console and Google Play will handle the rest. You can also use in-app updates to prompt users to update your app within your app.

How can I switch back to APK format from AAB format?

You cannot switch back to APK format from AAB format once you have published your app with AAB format. Google Play does not allow you to revert to a lower version code or a different publishing format. You can only switch to a higher version code or a newer publishing format.

e237b69de6